Oracle® Fusion Middleware

WebCenter Sites: Mobility Server Template API Guide

11*g* Release 1 (11.1.1.8.0)

E42080-01

July 2013

Oracle Fusion Middleware WebCenter Sites: Mobility Server Template API Guide, 11*g* Release 1 (11.1.1.8.0)

Table of Contents

Introduction

Oracle WebCenter Sites: Mobility Server, the newest deployment solution in web experience management, provides a single environment through which to create and manage dynamic content for thousands of mobile devices.

Mobility Server leverages the powerful CMS capabilities of Oracle WebCenter Sites beneath a remarkably easy to use interface. All traditional web content can be easily configured for use on myriad types of mobile devices, in real time.

The implementation guide describes Display Objects, the building blocks of Mobility Server mobile sites. Once a Display Object is mapped[1], a new template file can be created for it, overriding the original behavior. It is recommended that a copy be taken of the original Display Object template (found in the 'defaults' folder) as a boilerplate for further development.

[1] see the WebCenter Sites: Mobility Server Implementation Guide

MobileHelper API

For developers interested in further enhancing their mobile websites, this section describes the various Mobility Server "helper" functions that can be called from within templates. These functions allow the developer to access many of the Mobility Server properties programmatically and, where appropriate, allow the developer to pass in parameters to generate an appropriate and useful response from the server.

This section provides a list of these functions, along with a brief description and, where appropriate, parameters and code examples. The functions are grouped according to the following categories:

- Device Characteristics
- Property
- Images
- Layout
- Forms
- Maps
- Add-ons
- Video

Each function is available with either “camelCase” names (for example, “deviceHeight”, as shown in this document), or “c_style” names (for example, device_height is also a valid name for the same function). Developers can choose the function name format that fits their organization’s coding style.

Device Characteristics

deviceHeight

Description: Returns the device screen height in pixels. If the device can be rotated, the default orientation (portrait) is assumed. Note that this may include pixels used for browser address bars or other margins. See **maxImageHeight**() for calculating the max height for images.

Parameters: *None*

Example:

```
if ($h->deviceHeight() > 400) {
        // include CEO photo, plenty of screen real estate
}else {
        // display link to CEO photo only, want rest of content to appear above fold
}
```

deviceWidth

Description: Returns the device screen width in pixels. If the device can be rotated, the default orientation (portrait) is assumed. Note that this may include pixels used for browser scrollbars or other margins. See **maxImageWidth** for calculating the max width for images.

Parameters: *None*

Example:

```
if ($h->deviceWidth() > 300) {
        // include full message from CEO
}else {
        // display shorter message from CEO, this is not much screen to work with
}
```

maxImageHeight

Description: Returns the maximum image height in pixels that can be displayed for the client device. This function attempts to exclude the pixel height of browser margins (e.g. browser address bar). For actual device screen height in pixels, see **deviceHeight**.

Parameters: *$percentageValue*. Optional. Default value is 100. When this parameter is set, the maximum image height will be limited to this percentage of the screen. Accepts an integer representing a percentage value.

Example:

```
echo $h->thumbTag('/images/Article/1255732008194.jpg',
        $h->maxImageWidth(),
        $h->maxImageHeight(),
        80,
        array('alt'=>'My Image'));
```

Returns:

```
<img src="/cache/1255732008194-159x220.jpg" width="159" height="220"
alt="My Image">
```

maxImageWidth

Description: Returns the maximum displayable image width for the client device in pixels. This function attempts to exclude the pixel width of browser margins (e.g. scroll bars). If *$percentageValue* is set, it returns the percentage value for maximum image width.
For actual device screen width in pixels, see **deviceWidth**.
Parameters: *$percentageValue*. Optional. Default value is 100. When this parameter is set the maximum image width will be limited to this percentage of the screen. ~~A~~ccepts an integer representing a percentage value.

Example:

```
echo $h->thumbTag('/images/Article/1255732008194.jpg',
        $h->maxImageWidth(80),
        $h->maxImageHeight(),
        80,
        array('alt'=>'My Image'));
```

Returns:

```
<img src="/cache/1255732008194-159x220.jpg" width="159" height="220" alt="My Image"/>
```

supportsBackgroundColor

Description: Returns true if the device's browser supports the background-color CSS property. Mobile browsers that don't support background color can cause contrast problems; this function is used to avoid such issues (e.g. white-on-white).

Parameters: *None*

Example:

```
if($h->supportsBackgroundColor()){
        $background_color = '#CCC';
}
```

supportsFileUpload

Description: Returns true if the client device supports uploading files, false otherwise.

Parameters: *None*

Example:

```
if ($h->supportsFileUpload()) {
        echo 'Upload avatar: <input type="file" name="avatar"/>';
}
```

supportsLandscapeView

Description: Returns true if the device supports multiple orientations, false otherwise.

Parameters: *None*

Example:

```
if ($h->supportsLandscapeView()) {
		echo $h->getOrientedImage(...)
} else {
		echo $h->thumbTag(...)
}
```

isWebkitDevice

Description: Returns true if the mobile device is using a webkit-based browser, otherwise false.

Parameters: *None*

Example:

```
if ($h->isWebkitDevice()) {
		// some webkit-specific code
} else {
		// generic code
}
```

isAndroidDevice

Description: Returns true if the mobile device is identifying itself as an Android, otherwise false.

Parameters: *None*

Example:

```
if ($h->isAndroidDevice()) {
		// some Android-specific code
} else {
		// generic code
}
```

isIosDevice

Description: Returns true if the mobile device is identifying itself as an iOS device, otherwise false.

Parameters: *None*

Example:

```
if ($h->isIosDevice()) {
		// some iPhone/iPad-specific code
} else {
		// generic code
}
```

isWindowsPhoneDevice

Description: Returns true if the mobile device is identifying itself as a Windows phone, otherwise false.

Parameters: *None*

Example:

```
if ($h->isWindowsPhoneDevice()) {
		// some Window-Phone-specific code
} else {
		// generic code
}
```

Property

devicedbValueByKey

Description: Retrieves the appropriate value for the given device corresponding to a key in the device database, for example WURFL or DetectRight. A full and current list of keys can be found at each provider's website.

This function should only be used for device properties that are not supported by Mobility Server helper functions. Mobility Server helper functions often use additional logic that is either not represented in the device database or is specific to Mobility Server or pseudo-devices that are used by Mobility Server.

Returns boolean false when key was not found.

(Alias (deprecated): wurflValueByKey)

Parameters: $capabilitiesKeyName. The name of the property.

Example:

```
if ($h->wurflValueByKey('cookie_support') == 'true') {
        setcookie("hasVisitedBefore", true);
}
```

p

Description: Retrieves a Mobility Server property value corresponding to the given key.

Parameters:

$properties. *Optional*. An array of properties to use instead of current database contents.

$key. The name of the property. (See appendix for full list.)

$default. *Optional*. Default value is false.

Example:

```
if ($h->p(Constants::P_MAPS_VISIBLE) == 'true') {
        echo '<p>Click on one of the maps below for directions.</p>';
}
// This is identical to the above
if ($h->p('maps-visible') == 'true') {
        echo '<p>Click on one of the maps below for directions.</p>';
}
<p style="font-size: <?php echo $h->p(Constants::P_FONT_SIZE, 12); ?>px;">This is a paragraph with a font size. If no font size was defined, this paragraph will be in 12 pixel font, as per default.</p>
```

pVisible

Description: Convenience function, tests result of p($propertyName) against ‘visible’

Parameters:

$propertyKey. Name of property to test for being visible.

Images

imageAspectRatio

Description: Returns the aspect ratio of the image at *$path*, where *$path* is a URL relative to the web/ directory. The aspect ratio is computed by dividing the width of the image by the height of the image. Returns false when image is not found or file is not recognized as an image.

Parameters: $imagePath. The path to the image relative to the web directory.

Example:

```
// This image should be in MOBILITY_SERVER_FOLDER/web/images/Article/test.jpg
$aspectRatio = $h->imageAspectRatio('/images/Article/test.jpg');
echo $aspectRatio;
// 1.4802631578947, for example.
if ($aspectRatio > 1) {
        echo '<br/>'; // linebreak before landscape images
}
```

processInlineImages

Description: Returns a text block with all images optimized for use on mobile devices according to the given parameters. Each image encountered in the text block is converted into a thumbnail with given dimensions, or, if not specified, the maximum displayable on the given device. Note that this function may harm performance in non-production environments.

Parameters:

$text. String containing text blocks with img tags

$maxWidth. *Optional*. Maximum image width. Default value is **maxImageWidth**

$maxHeight. *Optional*. Maximum image height. Default value is **maxImageHeight**

$quality. *Optional*. Quality of the resulting image. Default is the global MobilityServer P_IMAGE_QUALITY property, or 80 if that is not set.

$options. *Optional*. An optional array of key=>value pairs containing HTML attributes to be appended to resulting image tags. (e.g. "alt" or "title")

$crop. *Optional*. Default is false. True makes center-croppped thumbnails of exactly the specified height and width.

Example:

```
echo $h->processInlineImages('Text before image <img src="/images/Article/test.jpg"/> and after', 100, 100);
// Would print Text before image <img src="/cache/test-100x100.jpg"/> and after
```

getStandardThumbTag

Description: Standard thumb generation function. Returned image size depends on "Maximum Image Width" property and quality is set by global "Image Quality" property.

Parameters:

$path. Relative path starting from web/ directory

$altText. *Optional.* Alt attribute for image

Example:

```
echo $h->getStandardThumbTag('/images/Article/test.jpg', 'My image')
// returns <img src="/cache/test-290x200.jpg" alt="My image"/>
```

getThumbTag

Description: Returns an image tag based on the *$path*, where *$path* is a relative filesystem path to 'web/'. Note that the thumbnail image will always maintain the original image's aspect ratio, possibly resulting in an image that is less wide than the given maxWidth value or less tall than the given maxHeight value.

Parameters:

$path. Relative path starting from web/ directory

$maxWidth. *Optional*. Default is 100. Maximum image width in pixels.

$maxHeight. *Optional*. Default is 100. Maximum image height in pixels.

$options. *Optional*. Array of additional HTML attributes, given as key-value-pairs, to be added to the image tag

$crop. *Optional*. Default is false. True makes center-cropped thumbnails of exactly the specified height and width.

$quality. *Optional.* Default is the Mobility Server global Image Quality property, or 80 if that is not set.

$addImagePopupOptions. *Optional.* Default is true. If true, allows the a larger-size popup image to be displayed, if Mobility Server’s global Image Popup Allowed property is set.

$forceType. *Optional.* Default is false. If not false, forces image thumbnails to be converted to the specified content type (currently supports only image/jpeg).

Example:

```
echo $h->getThumbTag('/images/myimage.gif',
            160,
            140,
            90,
            array('class'=>'article_thumb', 'alt'=>'My Image'))
// returns <img src="/cache/myimage-160-140.gif" class="article_thumb" alt="My Image"/>
```

thumbifizeImage

Description: Returns the path to a thumbnail version of the image given in *$path*. The aspect ratio of the original image will be maintained, resulting in a thumbnail that is no wider than the given *$maxWidth*, and no taller than the given *$maxHeight*.

If the $crop parameter is supplied and true, the thumbnail will be exactly *$maxWidth* x *$maxHeight*; aspect ratio is maintained by cropping from the sides in the longer direction.

Parameters:

$path. Relative path to applications web/ directoryfilesystem

$maxWidth. *Optional*. Default is 100. Maximum desired width in pixels.

$maxHeight. *Optional*. Default is 100. Maximum desired height in pixels.

$quality. *Optional*. Default is 90. Indicates image quality of the thumbnail.

$crop. *Optional*. Default is false. if true, use maxWidth and maxHeight as minimum sizes, and center-crops the resized image to return a thumbnail of exactly the specified size.

$forceType. *Optional.* Default is false. If not false, forces image thumbnails to be converted to the specified content type (currently supports only image/jpeg).

Example:

```
echo $h->thumbifizeImage('/images/Article/myimage.gif', 200, 150);
// returns  "/cache/myimage-200-150.gif"
```

includeThumbTag

Description: Convenience method, echos the thumb tag created by getThumbTag() onto standard output.

Parameters:

$path. Relative path starting from web/ directory

$maxWidth. *Optional*. Default is 100. Maximum image width in pixels.

$maxHeight. *Optional*. Default is 100. Maximum image height in pixels.

$options. *Optional*. Array of additional HTML attributes, given as key-value-pairs, to be added to the image tag

$crop. *Optional*. Default is false. True makes center-cropped thumbnails of exactly the specified height and width.

$addImagePopupOptions. *Optional.* Default is true. If true, allows the a larger-size popup image to be displayed, if Mobility Server's global Image Popup Allowed property is set.

$forceType. *Optional.* Default is false. If not false, forces image thumbnails to be converted to the specified content type (currently supports only image/jpeg).

includeStandardThumbTag

Description: Convenience method, echos a thumb tag with most options set to their defaults onto standard output.

Parameters:

$path. Relative path starting from web/ directory

$altText. Alt text for the resulting <image> tag.

$options. *Optional*. Array of additional HTML attributes, given as key-value-pairs, to be added to the image tag

Layout

includeDisplayObject

Description: embeds a DisplayObject.

Parameters:

$label. The Label of the mapped display object.

$parameter. *Optional in most cases*. Usually a CID. Exact meaning depends on the source_type column in the display_object_type mapping table.

$vars. *Optional.* Array of settings to be passed to the DO tempalte. Defaults to empty.

Example:

```
$h->includeDisplayObject("PromoCarousel");
$h->includeDisplayObject("ArticleList", '1255731768473');
```

getDetailLink

Description: Returns a link tag to the detail page for a particular display object, if a Detail display object is mapped for the asset.

Parameters:

$do. The Display Object.

$headline. The text of the link.

$options - Optional. Array of additional HTML attributes, given as key-value-pairs, to be added to the image tag.

Example:

```
foreach ($h->displayObjectList as $a) {
        echo $h->getDetailLink($a, "Read more!");
        echo '<br/>';
}
```

getDetailUrl

Description: Returns a URL for the detail page for a particular display object, if a Detail display object is mapped for the asset.

Parameters:

$do. The Display Object.

Example:

```
  foreach ($h->displayObjectList as $i => $a) {
    $detail_url = $h->get_detail_url($a);
    echo '<a class="arrow" href="' . $detail_url . '">'.$a->getTitle().’</a>’;
  }
```

setTitle

Description: Sets the HTML TITLE attribute of a page.

Parameters:

$title. The title text.

Example:

```
$h->setTitle("Welcome to MobilityServer!");
```

caption

Description: Returns a shortened version of the string *$text* if *$text* is longer than *$length* and adds '...' to the end of the string. Note that on supported devices the CSS3 text-overflow options should be considered instead.

Parameters:

$text. Text that may be shortened

$length. *Optional*. Default value is 200. Number of characters before $text is truncated

Example:

```
echo $h->caption("This is a very, very, very, very long string.",15);
// returns "This is a very,..."
```

hasItems

Description: Returns true when the list parameter contains an array with one or more items.

Parameters:

$list. Any array.

Example:

```
if (!$h->hasItems($blasts)) {
		echo "No blasts available.";
} else {
		//render list
}
```

javascriptLink

Description: Given some number of paths (at least one), will return the <link> tag(s) for those paths. If the last parameter, *$options*, is a key-value-pair array, those attributes will be appended to all links tags. All files get a cache-busting query string appended to them.

Parameters:

$path. A relative path from web/. This parameter may be repeated indefinitely.

$options. *Optional*. Array of additional HTML attributes, given as key-value-pairs, to be added to the tag

Example:

```
echo $h->javascriptLink('/js/smartphonebb/top.js');
// returns <script type="text/javascript" src="/js/smartphonebb/top.js?t=1292880937"></script>
echo $h->javascriptLink(
  '/jqtouch/jqtouch/jquery-1.4.2.min.js',
  '/jqtouch/jqtouch/jqtouch.ms.js',
  '/jqtouch/jqtouch/jqt.location.js'
);
// returns:
<script type="text/javascript" src="/jqtouch/jqtouch/jquery-1.4.2.min.js?t=1292880937"></script>
<script type="text/javascript" src="/jqtouch/jqtouch/jqtouch.ms.js?t=1292880937"></script>
<script type="text/javascript" src="/jqtouch/jqtouch/jqt.location.js?t=1292880937"></script>
```

linkToPhoneNumber

Description: Returns an anchor tag that can be clicked on in a mobile browser to bring up the device's dialpad. Possible number formats are:
12223334444
2223334444
+1 222 333 4444
1 222 333 4444
222-333-4444
222.333.4444

Parameters:

$number. The phone number to dial.

Example:

```
echo $h->linkToPhoneNumber('+1 222 333 4444');
// returns <a rel="external" href="tel:+12223334444">(222) 333-4444</a>
```

slugify

Description: Replaces all non-letters or digits with a "-", turns all of the text into lowercase, and trims any whitespace from the ends. Useful for creating SEO-friendly URLs.

Parameters:

$text. Some text to slugify.

Example:

```
echo $h->slugify("My Article Headline");
// returns "my-article-headline"
```

stylesheetLink

Description: Given some number of paths (at least one), will return the <link> tag(s) for those paths. If the last parameter is a key-value-pair array, those attributes will be appended to all link tags. All files get a cache-busting query string appended to them.

Parameters:

$path. A relative path from web/. This parameter may be repeated indefinitely.

$options. *Optional.* array of additional HTML attributes, given as key-value-pairs, to be added to the tag

Example:

```
echo $h->stylesheet_link(
   '/jqtouch/jqtouch/jqtouch.css',
   '/client/universaldemo/css/touch.css',
   '/client/universaldemo/css/carousel.css',
   '/jqtouch/themes/universaldemo/theme.css',
   array('media' => 'screen')
   );
// returns:
<link rel="stylesheet" type="text/css" media="screen" href="/jqtouch/jqtouch/jqtouch.css?t=1292880936" />
<link rel="stylesheet" type="text/css" media="screen" href="/client/universaldemo/css/touch.css?t=1292880936" />
<link rel="stylesheet" type="text/css" media="screen" href="/client/universaldemo/css/carousel.css?t=1292880937" />
<link rel="stylesheet" type="text/css" media="screen" href="/jqtouch/themes/universaldemo/theme.css?t=1292880937" />
```

formatDate

Description: Returns a string with a date formatted to match the "Date Format" Mobility Server property.

Parameters:

$dateString. String representing date value, e.g. 2001-08-12. Supports all formats used by PHP's strtotime function (http://php.net/strtotime).

$relativeTime. *Optional*. Default is false. If set to true, assumes that $dateString is relative to the current time.

Example:

```
echo $h->formatDate('last Monday');
// on Monday the 20th of December 2010, if the "Date Format" Mobility Server property is set to "02/05/2010", it would return "12/13/2010"
```

backgroundColor

Description: Generates a proper background: or background-image tag for the given property value. Note that the $backgroundImageUrl is used for menu & listing buttons which have both gradient and background images. Background-position settings are done in css.
Usually, the argument is simply the value of the background color property for the Display Object.

Parameters:

$colorString. *Optional.* String representing a color value (defaults to #ffffff); or “transparent” for a transparent background; or a gradient color definition (for example "#ffffff 0%|#cccccc 100%").

$direction. *Optional.* Direction of the gradient, “top” or “bottom”.

$backgroundImageUrl. *Optional.* URL for the background image, or false (default) for no image.

Example:

```
div.<?php $h->includeClassname(); ?> {
   margin: <?php echo $h->containerMargin(); ?>;
   text-align: <?php echo $h->p(Constants::P_CONTAINER_TEXT_ALIGN); ?>;
   <?php echo $h->backgroundColor($h->p(Constants::P_BACKGROUND_COLOR));?>
   color: <?php echo $h->p(Constants::P_FONT_COLOR); ?>;
   font-size: <?php echo $h->p(Constants::P_FONT_SIZE); ?>;
}
```

containerMargin

Description: Generates a CSS margin attribute using the Display Object's Container Margin values.

Parameters:

none.

Example:

```
div.<?php $h->includeClassname(); ?> {
   margin: <?php echo $h->containerMargin(); ?>;
   text-align: <?php echo $h->p(Constants::P_CONTAINER_TEXT_ALIGN); ?>;
}
```

getClassname

Description: Returns the name of the Display Object mapping. This is useful in defining CSS classes and so on, the distinguish between Display Object mappings on the same page.

Parameters:

none.

Example:

```
echo '<ul class="$h->getClassname() item">';
```

includeClassname

Description: Echos the name of the Display Object mapping onto standard output. This is useful in defining CSS classes and so on, the distinguish between Display Object mappings on the same page.

Parameters:

none.

Example:

```
  ul.<?php $h->includeClassname(); ?> {
  <?php echo $h->backgroundColor($h->p(Constants::P_BACKGROUND_COLOR), 'top');?>;
}
```

getRGBAValue

Description: Returns an RGBA value for a use as (e.g.) a background color.

Parameters:

$colorString. An RGB hex value (such as #CCCCCC), or “transparent”.

$opacity. *Optional.* The opacity percentage to use in the RGBA value. Default is false for none.

Example:

```
#popupImage {
  background: <?php
    echo $h->getRGBAValue(
      $h->p(Constants::P_POPUP_SCREEN_COLOR, '#000'),
      round($h->p(Constants::P_IMAGE_POPUP_BACKGROUND_OPACITY, 70)/100, 2))
  ?>;
}
```

includeHeadStyle

Description: Echos “headstyle” onto standard output; used as a marker in the <style> tags within Display Object templates that should be moved into the page’s <head> section.

Parameters:

none.

Example:

```
<style class="<?php $h->includeHeadStyle(); ?>">
```

includeLiveDoScript

Description: Inserts a script for live-updating content (whose URL is specified in the page request).

Parameters:

$updateInterval. *Optional.* Interval in milliseconds at which to update the content. Defaults to 30000 (30 seconds).

Example:

```
if ($h->p(Constants::P_LIVE_ENABLED) == Constants::P_VAL_TRUE) {
    $h->includeLiveDoScript();
  }
```

replaceEmbeddedAssets

Description: Includes Display Objects corresponding to assets embedded in another Display Object's attributes. You can embed assets in attributes in Oracle WebCenter Sites using the CKEditor. Then you map the embedded asset type in Mobility Server to a Display Object mapping with the name "Embedded" plus the asset type name, and use the replaceEmbeddedAssets method in your template to display the embedded DO mappings.

Parameters:

$field. The content of the Display Object containing the embedded assets.

Example:

```
		<div class="contentBody">
				<?php echo strip_tags($h->replaceEmbeddedAssets($h->displayObject->getBody()),
"<ul><li><a><img><u><b><i><br><p><em><strong><span><style><div>");
				?>
		</div>
```

Forms

alphaInputBox

Description: Returns the HTML code for an input box that, when selected, would bring up a text keyboard on the mobile device, or default user input to letters instead of numbers (this is usually the default). The resulting markup to accomplish this varies widely across devices.

Parameters:

$name. The name of the input form.

$options. *Optional*. Array of additional HTML attributes, given as key-value-pairs, to be added to the tag

Example:

```
echo $h->alphaInputBox('name');
// returns <input type="text" name="name" />
```

emailInputBox

Description: Returns the HTML code for an input box that, when selected, would bring up a keyboard optimized for entering email addresses or website URLs. If such a keyboard is not available for the given device, the output from **alphaInputBox** is returned instead.

Parameters:

$name. The name of the input form.

$options. *Optional*. Array of additional HTML attributes, given as key-value-pairs, to be added to the tag

Example:

```
echo $h->emailInputBox('contact');
// returns <input type="email" name="contact" /> for iPhone
```

numericInputBox

Description: Returns the HTML code for an input box that, when selected, would bring up a numeric keyboard, or default the user input type to numeric.

Parameters:

$name. The name of the input form.

$options. *Optional*. Array of additional HTML attributes, given as key-value-pairs, to be added to the tag

Example:

```
echo $h->numericInputBox('year);
// returns <input type="number" name="year" style="-wap-input-format: '*N';" />
```

Maps

mapThumb

Description: A wrapper for map thumbnail generators; currently supports Google Maps API. See **mapThumbGoogle** for more information.

mapThumbGoogle

Description: Returns HTML IMG tag holding the properly sized Google map centered to given longitude/latitude. See also **mapThumb**.

Parameters:

$lat. The latitude to center the map on.

$lon. The longitude to center the map on.

$size. *Optional*. Default is 300x100. Either a string in the AxB format, where A is width and B is height, or an array with two values, the first of which indicates width, and the second height.

$query. *Optional*. Additional query values to pass to Google Maps in the REST call.

$attributes. *Optional*. Array of additional HTML attributes, given as key-value-pairs, to be added to the image tag.

Example:

```
$h->mapThumbGoogle(40.755009, -73.992064, '280x180', array('sensor' => 'false'), array('alt'=>'Location of office'));
// returns <img src="http://maps.google.com/maps/api/staticmap?sensor=false&zoom=15&center=40.755009%2C-73.992064&size=280x180"
alt="Location of office"  />
```

mapLink

Description: A wrapper for map link generators; currently only supports google. See mapLinkGoogle for more information.

mapLinkGoogle

Description: Returns a link to Google Maps centered at *$address*, which can be any string that Google will recognize.

Parameters:

$text. The text to use for the link.

$address. The address.

$attributes. A key=>value pair array that will be appended to the link element.

Example:

```
echo $h->mapLink(
        'Click here for store map.',
        '34.0632277,-118.4151699',
        array(
                'rel' => 'external',
                'onclick' => "location.href='http://maps.google.com/maps?q=34.0632277,-118.4151699";return false;"
        )
);
// returns
<a href="http://maps.google.com/maps?q=34.0632277,-118.4151699" rel="external"
onclick="location.href='http://maps.google.com/maps?q=34.0632277,-118.4151699;return false;">Click here for store map.</a>
```

Display Object Fields

Inside a template, a display object is accessible via $h->displayObject, unless it is a Listing Display Object, in which case an array of Display Objects is stored in $h->displayObjectList.

To access a Display Object's mapped field in the template, use
$h->displayObject->get**{FieldLabel}**
where FieldLabel is the camel-case field name - meaning that any underscores have been removed, and the first character after each underscore has been capitalized.

Example:

Field Label	**Camel-Case Field Label**
content	Content
author_name	AuthorName
h1	H1
h2_tag	H2Tag

For instance, if you have mapped the WebCenter Sites asset type "Articles" to a Detail Display Object, and mapped the fields as follows:

WebCenter Sites Field	**Field Label**
Author	Author
Title	Headline
Copy	Body

you could access the fields as follows:

```
echo '<h1>' . $h->displayObject->getHeadline() . '</h1>';
echo 'by ' . $h->displayObject->getAuthor();
echo '<p>' . $h->displayObject->getBody() . '</p>';
```

For a Listing Display Object, you must loop through the array in order to access the individual assets.

Consider the following mapping of ArticleDetail onto a Listing Display Object, with the same fields as above:

```
echo '<ul>';
foreach ($h->displayObjectList as $a) { ?>
		echo '<li>' . $a->getHeadline() . ' - by ' . $a->getAuthor() . '</li>';
}
echo '</ul>';
```

Asset Associations

Asset associations are brought over from WebCenter Sites without any need to map them - all that is required to use them is knowing the names.

For instance, if the Articles asset type has an association named RelatedNews, and RelatedNews contained assets that were mapped to a Detail Display Object with the label RelatedBlurb and just one field:

WebCenter Sites Field	Field Label
Title	Headline

You could list them like so:

```
echo '<h1>Related Articles</h1>';
foreach($h->displayObject->getAssociationAssets('RelatedNews') as $asset) {
        $h->include_display_object('RelatedBlurb', $asset->getCid());
}
```

with the templates/{family}/RelatedBlurb.php containing:

```
echo '<br/><strong>Related Story:</strong> ' . $h->displayObject->getHeadline();
```

Warning: Associations are currently only supported for non-Listing Display Objects.

Asset XML

Inside a template, the XML for the asset associated with a display object is accessible via $h->displayObject->getAssetXml. This enables a template to access properties of an asset that are not mapped into the display object itself.

To access a Display Object's asset XML, use
$h->displayObject->getAssetXml()
which returns a SimpleXmlElement object (see the PHP manual for the description and API of this object).

Appendix 1: List of Available Properties

This is a partial list of available properties. (See the p function.) This list will grow as additional Display Objects are installed.

P_ADS_VISIBLE
P_BACKGROUND_COLOR
P_BASIC_NAVIGATION_STYLE
P_BORDER_COLOR
P_BORDER_STYLE
P_BORDER_VISIBLE
P_CAPTION_FONT_COLOR
P_CAPTION_FONT_SIZE
P_CAPTION_LENGTH
P_CAPTION_VISIBLE
P_COLLAPSIBLE_ANIMATION
P_COLLAPSIBLE_BORDER_COLOR
P_COLLAPSIBLE_CONTENT_BG
P_COLLAPSIBLE_TITLE_ACTIVE_BG
P_COLLAPSIBLE_TITLE_ACTIVE_FONT_COLOR
P_COLLAPSIBLE_TITLE_BG
P_COLLAPSIBLE_TITLE_FONT_COLOR
P_COLLAPSIBLE_TITLE_FONT_SIZE
P_COLLAPSIBLE_TITLE_TEXT_ALIGN
P_COLLAPSIBLE_TITLE_WRAP_STYLE
P_CONTAINER_MARGIN_BOTTOM
P_CONTAINER_MARGIN_LEFT
P_CONTAINER_MARGIN_RIGHT
P_CONTAINER_MARGIN_TOP
P_CONTAINER_TEXT_ALIGN
P_CROP_IMAGES
P_DATE_FORMAT
P_EMAIL_BUTTON_VISIBLE
P_FACEBOOK_BUTTON_VISIBLE
P_FONT_COLOR
P_FONT_CONTROL_BACKGROUND
P_FONT_CONTROL_BORDER_COLOR

P_FONT_CONTROL_POSITION

P_FONT_FAMILY

P_FONT_SIZE

P_FOOTER_BACKGROUND_COLOR

P_GRIDLIST_TEXT_POSITION

P_HEADLINE_FONT_COLOR

P_HEADLINE_FONT_SIZE

P_HEADLINE_LENGTH

P_HEADLINE_LOCATION

P_HEADLINE_VISIBLE

P_IMAGES_VISIBLE

P_IMAGE_ALIGN

P_IMAGE_CONVERSION

P_IMAGE_POPUP_ALLOWED

P_IMAGE_POPUP_BACKGROUND_OPACITY

P_IMAGE_POPUP_ICON_VISIBILITY

P_IMAGE_QUALITY

P_INDICATOR_FONT_SIZE

P_INDICATOR_OFF_COLOR

P_INDICATOR_ON_COLOR

P_INDICATOR_VISIBLE

P_LISTING_FEATURED_IMAGE_LOCATION

P_LISTING_FEATURED_MAP_LOCATION

P_LISTING_IMAGE_LOCATION

P_LISTING_MAP_LOCATION

P_LIST_ITEM_BACKGROUND_COLOR

P_LIVE_ENABLED

P_LOADING_SCREEN_OPACITY

P_LOGO_BACKGROUND_COLOR

P_MAPS_VISIBLE

P_MAP_ALIGNMENT

P_MAP_VISIBLE

P_MAP_WIDTH

P_MAX_IMAGE_WIDTH

P_MAX_LISTING_FEATURED_IMAGE_WIDTH

P_MAX_LISTING_FEATURED_MAP_WIDTH

P_MAX_LISTING_IMAGE_WIDTH

P_MAX_LISTING_MAP_WIDTH

P_MENU_BUTTON_ACTIVE_COLOR

P_MENU_BUTTON_BACKGROUND_COLOR

P_MENU_BUTTON_FOREGROUND_COLOR

P_MENU_OPENING_STYLE

P_MENU_SEPARATOR_COLOR
P_NAV_BUTTON_ALIGN
P_PINTEREST_BUTTON_VISIBLE
P_POPUP_SCREEN_COLOR
P_POST_TAP_BACKGROUND_COLOR
P_POST_TAP_FONT_COLOR
P_PRICE_FONT_COLOR
P_PRICE_FONT_SIZE
P_SHARE_AND_FONT_BUTTONS_BACKGROUND_COLOR
P_SMART_NAVIGATION_STYLE
P_SOCIAL_MEDIA_BUTTON_ICON_THEME
P_TABGROUP_CONTENT_BG
P_TABGROUP_TAB_ACTIVE_BG
P_TABGROUP_TAB_ACTIVE_FONT_COLOR
P_TABGROUP_TAB_BG
P_TABGROUP_TAB_CORNER_STYLE
P_TABGROUP_TAB_FONT_COLOR
P_TABGROUP_TAB_FONT_SIZE
P_TABGROUP_TAB_TEXT_ALIGN
P_TABGROUP_TAB_WRAP_STYLE
P_TABLE_WIDTH
P_TEMPERATURE_TYPE
P_TOUCH_NAVIGATION_STYLE
P_TRANSITION_STYLE
P_TWITTER_BUTTON_VISIBLE